Optimización de parámetros para el problema del viajante utilizando un algoritmo de colonia de hormigas

Alejandro Francisco Toral[1]

[1] Escuela Técnica Superior de Ingenieros Informático, Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Madrid
a.ftoral@alumnos.upm.es

Resumen. El problema del viajante, más conocido como Traveling Salesman Problem (TSP), es un popular problema utilizado para experimentar con algoritmos de optimización. En este trabajo se intentarán resolver una serie de problemas ya definidos[3] con un algoritmo de colonia de hormigas (Ant Colony Algorithm, ACA). Para encontrar la solución óptima se aplicarán cambios en todos los parámetros involucrados en el sistema y se procederá al análisis y comparación de los resultados obtenidos.

Keywords: sistemas multiagente, colonia de hormigas, problema del viajante.

1 Introducción

1.1 Problema del viajante

El problema del viajante (TSP) consiste en encontrar el camino más corto a la hora visitar todas las ciudades, de modo que el nodo de comienzo y de fin sean el mismo punto, y el resto de nodos sean visitados una sola vez por viaje.

Los mejores resultados del TSP se obtienen rápidamente con grafos de pocas ciudades (de 10 a 100), cuando en situaciones de la vida cotidiana este problema se aplica en situaciones de cientos y miles de ciudades. Para esto, se suelen utilizar algoritmos evolutivos, inspirados en procesos naturales, como los algoritmos genéticos, los enjambres de partículas o los algoritmos de colonias de hormigas. Estos procesos están enfocados en una optimización global del problema mediante la repetición y la creación de nuevas generaciones capaces de mejorar los resultados ya obtenidos.

1.2 Algoritmo de la colonia de hormigas

Este algoritmo se trata de una técnica probabilística para solucionar problemas computacionales que pueden reducirse a buscar los mejores caminos en grafos. Se basa en el comportamiento de una colonia de hormigas: con la ayuda de feromonas (sustancias químicas segregadas), las hormigas marcan sus pasos y se ayudan entre ellas a encontrar un buen camino para llegar a un objetivo común.

Para su aplicación en el problema del viajante, tendremos como elemento de entrada un grafo $G(V, E)$, donde V son los nodos y E, el set de aristas que conectan dichos

nodos. Cada una de estas aristas tendrá a su vez una distancia d_{ij} y una cantidad τ_{ij} de feromonas.

Probabilidad de cambio. Las hormigas se inicializan aleatoriamente en cualquiera de los puntos del grafo. Después, para moverse, siguen una probabilidad p, definida por la siguiente fórmula:

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha} \cdot (\eta_{ij})^{\beta}}{\sum_{g} (\tau_{ig})^{\alpha} \cdot (\eta_{ig})^{\beta}} \text{ sí } j \text{ no se ha visitado aún,}$$
$$p_{ij}^{k} = 0 \text{ si ya se ha visitado ese nodo;}$$

donde los parámetros α y β son constantes utilizadas para controlar el peso de las feromonas y la heurística η_{ij} es la inversa de las distancias d_{ij}.

Actualización de las feromonas. El valor de las feromonas cambia según el tiempo que pase. Todas las hormigas contribuyen a actualizar las cantidades de feromonas en cada borde de la siguiente manera:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{k},$$

donde ρ ($0 \leq \rho \leq 1$), es la ratio de evaporación de las feromonas, se usa para evitar la acumulación ilimitada de los rastros de feromonas y permite que el algoritmo olvide las malas decisiones tomadas previamente; *m* es el número de hormigas de la colonia y el factor $\Delta\tau_{ij}^{k}$ es la cantidad de feromonas depositada en el camino por una hormiga *k* en el nodo (i, j) de tal manera que:

$$\Delta\tau_{ij}^{k} = Q/L_{best} \text{ si la hormiga } k \text{ visita el nodo } (i, j),$$
$$\Delta\tau_{ij}^{k} = 0 \text{ si no;}$$

donde Q es el factor de cambio de las feromonas, una constante; y L_{best} es la longitud del mejor tour que una hormiga de la colonia ha hecho en cada iteración.

2 Diseño del algoritmo

El modelo está basado en el ACO básico proporcionado en la presentación "*Using Collective Intelligence for Search Optimization*" de Nik Swoboda[1], así como en el algoritmo de optimización de la colonia de hormigas de la librería scikit-opt[2].

Primero se inicializan los parámetros que se van a utilizar a lo largo del programa: número de nodos; feromonas iniciales (*tau*, calculadas como m/C^{nn} , con *m* como el número de hormigas y C^{nn} como la duración del recorrido generado por la heurística del vecino más cercano; matriz de distancias euclídeas; y la matriz de distancias inversas (*eta*).

Para determinar la condición de parada del bucle principal se establece un número de iteraciones a cumplir. Dentro del bucle principal se genera una tabla donde se registran los movimientos de las hormigas de cada iteración. Cada hormiga comienza su recorrido en un nodo al azar del grafo, y continúa su camino siguiendo la probabilidad p_{ij}^k definida anteriormente, evitando repetir ciudades. Al terminar, la tabla estará completa con los caminos de cada hormiga de la colonia en una iteración. Una vez rellenada, se busca el mejor de los caminos mediante el cálculo de la distancia total de cada uno de ellos. Una vez se adquiere, se añade a la lista de mejores caminos, se actualizan las feromonas y se repite el proceso hasta terminar las iteraciones.

El resultado de este bucle es una lista con el mejor camino de cada iteración. Para terminar, se vuelve a calcular la distancia para cada camino, esta vez teniendo en cuenta la lista de los mejores, y se devuelve aquel que corresponda con la menor distancia recorrida.

3 Experimentos

El ajuste de parámetros se realizó con el set de datos de menor tamaño: *dj.tsp*, un dataset de 38 ciudades de Djibouti, cuyo objetivo es conseguir una distancia recorrida mínima de 6656. Estos experimentos han sentado varias bases que se discutirán a continuación y que reducirán la búsqueda de parámetros más sencilla, debido a la gran acotación que suponen estas primeras pruebas.

Se han ido cambiando de uno en uno una serie de parámetros clave del programa:

- Colonia (*m*): número total hormigas. Se incrementaron en múltiplos de 5. También se experimentó con *m* = número de nodos.
- Alpha (α). Varió en un intervalo de 0 a 2, con incrementos de 0,5.
- Beta (β). Varió en intervalos de 0 a 6, con incrementos de 1.
- Q: se probó con un rango de 0 a 2, con incrementos de 0,5; y se aumentó excesivamente a 10.
- Rho (ρ): se probó con un rango de 0 a 0.9, con intervalos de 0.1.

El número de iteraciones se ha fijado en 100. En la tabla 1 se encuentran los resultados (distancia, D, y tiempo, T, en segundos) de este primer experimento.

Table 1. Estudio comparativo del primer dataset: *dj.tsp*

m	α	β	Q	ρ	D	T (s)
5	0.0	0.0	0.0	0.0	27.428	4.897897
10	0.0	0.0	0.0	0.0	27.237	8.976035
20	0.0	0.0	0.0	0.0	25.432	18.0258
	0.5	0.0	0.0	0.0	27.595	12.31106
	1.0	0.0	0.0	0.0	26.726	19.20172
		0.5	0.0	0.0	23.172	17.69367
		1.0	0.0	0.0	18.697	14.54810
		1.5	0.0	0.0	14.912	15.56636

		2.0	0.0	0.0	11.658	16.88436
		3.0	0.0	0.0	10.147	17.35564
		4.5	0.0	0.0	8.815	15.38195
		6.0	0.0	0.0	8.555	13.44804
			0.5	0.0	8.470	16.0430
			1.0	0.0	8.412	15.65212
				0.1	8.386	17.81833
				0.5	8.533	17.80936
			1.5	0.0	8.538	13.51387
	1.5	0.0	0.0	0.0	27.120	16.09692
38	0.0	0.0	0.0	0.0	25.509	29.18795
	0.5	0.0	0.0	0.0	26.946	31.21351
	1.0	0.0	0.0	0.0	26.295	23.68068
	1.5	0.0	0.0	0.0	25.991	29.54402
		0.5	0.0	0.0	20.743	31.02889
		1.0	0.0	0.0	17.344	32.84920
		2.0	0.0	0.0	11.440	34.60987
		3.0	0.0	0.0	9.683	29.91201
		4.5	0.0	0.0	8.688	31.04602
		6.0	0.0	0.0	8.468	31.36518
			0.5	0.0	8.386	27.50445
				0.1	8.457	25.35815
			1.0	0.0	8.447	28.50979
	2.0	0.0	0.0	0.0	27.059	30.60416
50	0.0	0.0	0.0	0.0	26.127	50.54925
	1.5	6.0	0.5	0.1	8.412	49.52307

4 Discusión de los resultados

El mejor resultado y tiempo se ha obtenido con una colonia de 20 hormigas y con los valores de $\alpha = 1.0$; $\beta = 6.0$; $Q = 1.0$ y $\rho = 0.1$, aunque con un número de hormigas igual al número de nodos, los resultados son iguales.

El resto de pruebas se han realizado con una serie de archivos TSP[3] de distinto número de nodos cada uno (desde las 38 ciudades hasta las 662).

Colonia

En las primeras ejecuciones, cuando el resto de parámetros se encontraban a 0, a medida que se aumentaba el número de hormigas, disminuía la distancia. Esto es debido a la cantidad de información recibida entre las hormigas de la colonia: a más hormigas, más caminos recorridos. Sin embargo, aumentar la colonia excesivamente sin ajustar el resto de parámetros, provoca un aumento de la distancia. Por otra parte, si el número de hormigas es superior al número de nodos y utilizamos los parámetros con los que mejores resultados se han obtenido posteriormente, se observa que el resultado apenas varía con

respecto del mejor, dando a entender que el camino no se puede mejorar más por muchas hormigas que haya en la colonia.

El número total de hormigas también influye en el tiempo de ejecución. Cuanto más grande es la colonia, más tiempo tarda en obtener resultados.

Parámetros α y β

Por lo que se ha podido observar a lo largo del experimento, el papel de los parámetros α y β es el siguiente: si $\alpha = 0$, solo se considera la heurística del problema, por lo que el algoritmo proporciona resultados propios de un algoritmo *greedy* aleatorio con múltiples posiciones iniciales. Si $\beta = 0$, se observa cómo se produce el fenómeno de explotación, donde las hormigas continúan todas por el mismo rastro de feromonas, sin hacer caso a ninguna heurística y constituyendo siempre la misma solución. Resultados parecidos se observan con valores de $\alpha > 1$.

Parámetros Q y ρ

El ajuste de los dos últimos parámetros resulta anecdótico, pues no suponen un cambio muy grande con respecto a anteriores resultados. Aun así, ayudan a acercar más la solución óptima del problema.

El cambio más destacable se obtiene al aumentar excesivamente el valor de la tasa de evaporación de feromonas, ρ. Para $\rho \geq 0.5$, las distancias recorridas por las hormigas vuelven a aumentar.

5 Conclusiones

Utilizando los parámetros ajustados de la fase de experimentación, y usando el número de nodos como número de hormigas de la colonia, los resultados fueron parecidos a los de la fase de experimentación: con el set de nodos de 131 ciudades, de óptimo 564, el mejor resultado ha sido 736; para 411, de un óptimo de 1343, el mejor resultado ha sido de 1882…

A pesar de no llegar a los resultados óptimos para ninguno de los caminos probados, el algoritmo hace un buen trabajo ya que se acerca bastante a dichos números (en torno al 12% por encima del óptimo) y el tiempo de respuesta es bastante bajo en comparación con otros algoritmos y librerías ya existentes.

Como futuro trabajo se podría considerar el ajuste de parámetros automático y la optimización –aún mayor– del tiempo, mediante la eliminación de bucles o el uso de diferentes heurísticas distintas de la distancia euclídea.

6 Referencias

1. Using Collective Intelligence for Search Optimization. Swoboda, Nikolaus Guyon. 2020
2. https://pypi.org/project/scikit-opt/
3. http://www.math.uwaterloo.ca/tsp/vlsi/index.html